

Batch Script - Debugging

Debugging a batch script becomes important when you are working on a big complex batch script.

Following are the ways in which you can debug the batch file.

Using echo command

A very simple debug option is to make use of echo command in your batch script wherever possible. It will display the message in the command prompt and help you debug where things have gone wrong.

Here is a simple example that displays even numbers based on the input given. The echo command is used to display the result and also if the input is not given. Similarly, the echo command can be used in place when you think that the error can happen. For example, if the input given is a negative number, less than 2, etc.

Example

```
@echo off
if [%1] == [] (
    echo input value not provided
    goto stop
)
rem Display numbers
for /l %%n in (2,2,%1) do (
    echo %%n
)
:stop
pause
```

Output

```
C:\>test.bat
10
2
4
6
8
10
```

22

Press any key to continue ...

Using pause command

Another way is to pause the batch execution when there is an error. When the script is paused, the developer can fix the issue and restart the processing.

In the example below, the batch script is paused as the input value is mandatory and not provided.

Example

```
@echo off
if [%1] == [] (
    echo input value not provided
    goto stop
) else (
    echo "Valid value"
)
:stop
pause
```

Output

```
C:\>test.bat
input value not provided
Press any key to continue..
```

Logging the error messages to another file

It might get hard to debug the error just looking at a bunch of echo displayed on the command prompt. Another easy way out is to log those messages in another file and view it step by step to understand what went wrong.

Here is an example, consider the following test.bat file:

```
net statistics /Server
```

The command given in the .bat file is wrong. Let us log the message and see what we get.

Execute the following command in your command line:

```
C:\>test.bat > testlog.txt 2> testerrors.txt
```

The file testerrors.txt will display the error messages as shown below:

```
The option /SERVER is unknown.
The syntax of this command is:
NET STATISTICS
[WORKSTATION | SERVER]
More help is available by typing NET HELPMSG 3506.
```

Looking at the above file the developer can fix the program and execute again.

Using ErrorLevel to detect errors and log them

Errorlevel returns 0 if the command executes successfully and 1 if it fails.

Consider the following example:

```
@echo off
PING google.com
if errorlevel 1 GOTO stop
:stop
    echo Unable to connect to google.com
pause
```

During execution, you can see errors as well as logs:

```
C:\>test.bat > testlog.txt
```

testlog.txt

```
Pinging google.com [172.217.26.238] with 32 bytes of data:
Reply from 172.217.26.238: bytes=32 time=160ms TTL=111
Reply from 172.217.26.238: bytes=32 time=82ms TTL=111
Reply from 172.217.26.238: bytes=32 time=121ms TTL=111
Reply from 172.217.26.238: bytes=32 time=108ms TTL=111
Ping statistics for 172.217.26.238:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 82ms, Maximum = 160ms, Average = 117ms
    Connected successfully
Press any key to continue . . .
```

In case of failure, you will see the following logs inside testlog.txt.

Ping request could not find host google.com. Please check the name and try again.
Unable to connect to google.com
Press any key to continue . . .